In less than one second, we were able to evaluate a list of 10,000 passwords and discover which one was used in the hash algorithm.

With the Password and Hash function now known, we can begin our attempts to guess the remaining data set. While hash functions are normally one way, as long as we know some basic meta information about the dataset it becomes possible to decode other attributes as it's much easier to guess the values of classifiers than it is to guess the salt for a hash.

In the case of ZVirus classification, we know that the CDC has been labeling their publicly available statistics and graphs with Carrier, Immune and Infected. So it's likely that these three values were used in this dataset especially as it contains three unique classifiers in the ZVirus column. Of course it's possible they used non-contextual based classifiers such as 0, 1, 2, etc. Or even used partial contextual classifiers like C, Im, and In. As always, we started with the obvious and worked our way down. Using our found salt and a list of possible plaintext values, we were able to easily identify which hash value resulted from which classifier.

Each other feature follows the same path, taking educated guesses as to the values present, and running them against the given hashes. Once a pattern is identified it becomes even easier. For example, the hardest feature to guess is probably the Ethnicity/Race as those terms are somewhat inconsistent across all reporters, but if you knew the CDC used the census classifications (as they should) then it becomes all the easier to crack.

But let’s step back and assume that Bob couldn't crack the password and that whoever perturbed the data used a cryptographically strong method. Let's take a look at what he could do instead.

**Metacontextual Analysis**

Taking another look at our dataset, in particular hair and eye color, we noticed a strange coincidence. Within this dataset, because all values for all features were applied the same salts, any features that use the same values will have the same hashes. In this case, a particular hash is repeated for both hair color and eye color.

Going through a list of hair colors and eye colors, very few match up naturally on a person. Realistically speaking, the only color that appears in both hair and eyes (using common parlance) is brown in some fashion. Whether it's light brown, dark brown, or just brown, we can assume that this particular hash probably means brown in some way. This is an example of inferring meaning based on context and using any metaknowledge known about the dataset. Without ever having to know what that hash translates directly into we still have transformed it from incomprehensible into a usable bit of information. Similarly, it’s possible to decode the ethnicity feature column as well.

Now, Bob, like us, has access to the US Census data and would know that there's not just three ethnicities or races as our anonymized dataset might suggest. A quick search of our dataset gives us the following percentages for each unique hash string.

41a0446… 77.7%  
239bfd6… 14.2%  
f2c2851… 8.1%

Either the data was prepared in such a way that ethnicities were grouped together or only the most common ethnicities were selected for representation. While it's possible that certain groups were combined, such as maybe Native Americans and other indigenous people, it's unlikely that, say, White Americans and Black Americans were grouped together. If Bob knows that White Americans make up around 73% of the population, Black Americans about 13%, and Asian Americans around 8%, those values are close enough that Bob could in fact assume that those hashes equate in some way to those demographic populations.

This demonstrates a very important aspect to perturbance. Even with encrypted values, analysis can still be performed to assume a value or values for a feature if the feature is known and has associated data that’s available to the public. In this case a quick search of census data gives us near equivalent demographic statistics which makes it easy to draw a conclusion. We can easily perform this same analysis to uncover the meanings for the Hispanic\_Latino feature set as well by looking at the relative population of each value to the presumed White American hash.

Not every feature will be able to be inferred in this fashion. Bloodtype will (based on how we built our dataset), but for features like Sex or State, simple probability or statistical demographic analysis probably won't be enough. However, much like the popular number game Sudoku, once you start filling in the easy blanks, the harder blanks will follow. For example, we can verify our Ethnicity selections by looking at Hair and Eye color demographics and comparing them to our assumptions. Further, this is based entirely on this dataset alone. If Bob has access to any additional data, it just gets easier and easier to make those metacontextual connections necessary to identify feature values.

**Stopping Metacontextual Analysis**

Interestingly, after playing with the data, Age Class appeared to be the most difficult to predict and crack. Because the groupings didn't follow census data, but rather an internal logic, it would have been almost impossible for bob to put together which class/group equated to which hash, or even how to group ages together to begin with.

Certainly census data would be easy enough to deanonymize, but the only reason it worked so well with this dataset is because our original dataset proportionally represented a real population. Most datasets of this nature probably won't actually represent a real population, but it's important to remember for those datasets that do, especially for an extremely large number of observations. And in actuality, the easiest data to deanonymize were for features that had easily predictable values.

## Conclusion

It should be said that there is no "either K-Anonymization or Perturbance" (although this is "always remove PII"). Both are methods of anonymizing datasets and both are equally valuable in hindering malicious analysis. And while it would be an easy conclusion to say that it all depends on the dataset as to which methods should be employed, that's ultimately the predicated conclusion. Every dataset has a purpose, features and qualities that would make some methods more viable than others. In the case of perturbance, a purely quantitative and dense dataset would probably not be a very good candidate for most forms of perturbance - especially if precision is important to the final model. Each method has its own weaknesses and strengths which should be considered whenever anonymization is performed on a dataset.

**K-Anonymization**

Much of the strength behind K-Anonymization lies behind the principle of finding a needle in a needle-stack. Small datasets by their nature are susceptible to de-anonymization because there so few individuals that have been captured, that distinguishing between them becomes relatively simple. Differences, again, are not necessarily between individual features, but combinations of features as well. K-Anonymization can be employed to overcome this weakness by identifying those more unique individuals and removing them from the final dataset. This comes at a cost, however, as by removing individuals and observations from a dataset ultimately impacts the accuracy, precision and effectiveness of any models derived. But a sufficiently large dataset or a dataset with low variance would be perfect candidates for K-Anonymization without impacting the overall performance of any algorithms run on the dataset.

**Perturbance**

While machines have very little problem using values that are 64 characters long and anonymous feature names to develop models for classification or prediction, it can lead to an issue of readability for the human aspect of the team. A truly anonymous set might in fact be a list of features x1, x2, x3... with a collection of values that have been encrypted or fuzzed to such a degree that any attempt to apply any discovered models would be largely theoretical and probably practically useless. Any attempt a perturbance then weighs readibility and usability against confidentiality and anonymity. Largely, most concerns about the weaknesses of perturbance come down to the concerns related to most forms of data security. For example, using a proper salt or some form of cryptographically secure methods of encrypting/perturbing the data is a must. Whether or not to perturb and which methods to employ then depend on what is wanted out of a shared dataset - whether contributers are expected publish and interpret their own results or simply return them to the owners of the dataset; or whether confidentiality or precision are more important to the researchers.

It is not reasonable to avoid data collection and sharing altogether. Thus, we have compiled a series of recommendations that we would advise to those interested in having a better realization of the remaining vulnerability of their anonymization techniques.

1. Actively attempt to de-anonymize data sets. Depending on the sensitivity of the data, this might need to be performed by individuals not privvy to the anonymization processes used.
2. Treat normally innocuous data as weaknesses for an attacker to exploit.
3. Consider using multiple anonymization techniques. Test along the way to see how this affects statistical analysis.
4. Work to understand the goal of a data mining activity before providing data sets. Consider using subsets, and weigh the the value of data mining activities with the vulnerability of the subjects‘ information.
5. Consider the long range goals of an organization before employing the use of public contests. Will you even need this winning algorithm or model in the long term?